PANSAT FUNCTIONAL TESTING SOFTWARE AND SUPPORT HARDWARE

Jim A. Horning
Computer Engineer, Naval Postgraduate School

ABSTRACT

The Petite Amateur Navy Satellite (PANSAT) was successfully launched aboard the STS-95 *Discovery*. PANSAT provides digital store-and-forward communications in a low-Earth orbit, and is a platform for an instructional laboratory. The spacecraft was built and tested at the U.S. Naval Postgraduate School in Monterey, California, by officer students, research staff and faculty. The PANSAT digital hardware was prototyped with various commercial-off-the-shelf (COTS) hardware and software to assist in designing the unique systems which were eventually built and integrated into the spacecraft. This paper presents the hardware and software components used to develop the PANSAT spacecraft. Details of the evolution of design and testing are explained as well as the tools to complete the final functional verification performed during payload integration. The concepts presented are applicable to generic space flight experiments.

INTRODUCTION

PANSAT is a small satellite for digital store-and-forward communications in the amateur frequency band. It features a direct sequence spread spectrum differentially coded, binary phase shift keyed (BPSK) communication system at an operating frequency of 436.5 MHz. The store-and-forward capability will allow NPS and amateur radio operators to send or receive messages during several short communication windows every day.

The PANSAT project began in 1989 as an educational program for students at the Naval Postgraduate School's (NPS) Space Systems Academic Group (SSAG). The project goal is to provide meaningful and realistic research topics for students in the area of space systems engineering and space systems operations. In doing so, the Space Systems Academic Group has prepared students for space related tasks and has developed an infrastructure of facilities and personnel capable of developing space qualified systems.

The entire PANSAT structure weighs approximately 125 pounds, has a diameter of about 19 inches, and was designed to be launched as a secondary payload from the Space Shuttle via the Hitchhiker Program. PANSAT is a 26-sided polyhedron, as shown in Figure 1 with some of the side panels removed in order to show the interior, a configuration chosen to narrow the range of solar flux because the spacecraft tumbles freely. PANSAT is not stabilized and tumbles freely. The satellite uses an omni-directional antenna system consisting of four quarter-wave-length segments to achieve near uniform signal coverage regardless of PANSAT's orientation.

The design, fabrication, testing, and integration of the entire spacecraft were performed at the Space Systems Academic Group at the Naval Postgraduate School. One of the early design objectives was the ability to easily perform functional testing, starting with the prototype subsystems and extending all the way to the system level integration. This was accomplished by reusing many software test components. The support hardware consisted mainly of a general purpose PC and a power supply. Final integration testing performed at NASA Goddard Space Flight Center (GSFC) required only two laptop computers, a power supply, and a brief-case sized RF/Modem unit.

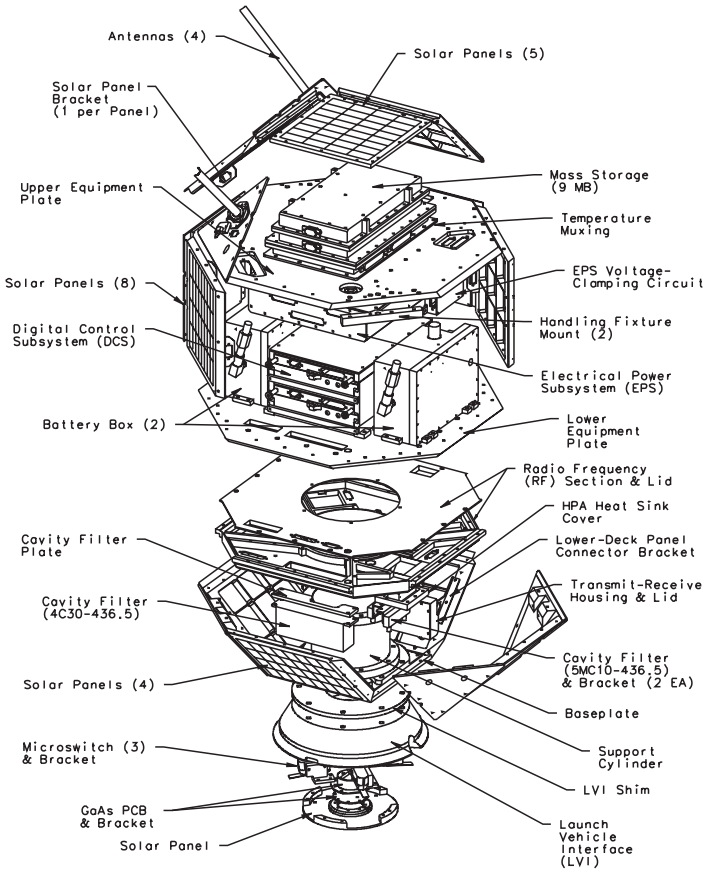

Figure 1. PANSAT Expanded View.

PANSAT ELECTRONIC MODULES ARCHITECTURE

PANSAT electronic modules, as depicted in Figure 2, consist of various subsystems: digital control system (DCS) which includes a computer and the modem, analog temperature multiplexers (TMUX), mass storage (MS) for data and message storage, radio frequency (RF) system for up and down conversion of the communication signals, and the electrical power system (EPS) for the distribution of power on the satellite. Two DCS modules provide a fault tolerant computer and modem pair. Only one DCS is powered and operating at any time.

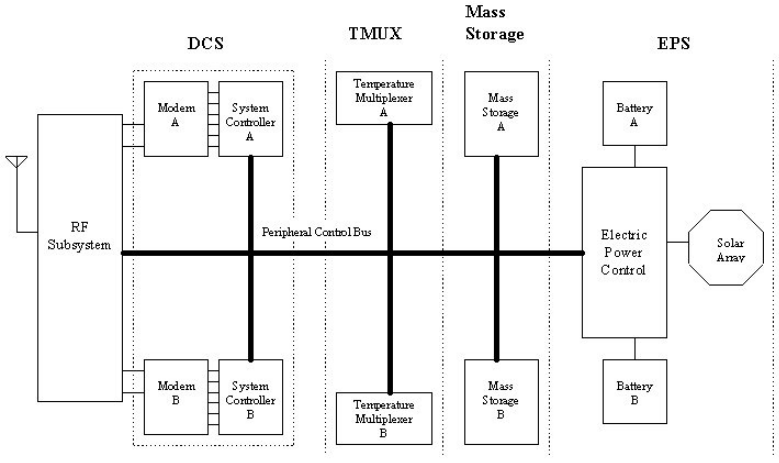

Figure 2. PANSAT Electronic Modules Block Diagram.

Connecting The Electronic Modules Via The Peripheral Control Bus

The PANSAT electronic modules are all interconnected by the peripheral control bus (PCB). The PCB is a power, control, and eight-bit parallel data bus where each module is assigned a particular address. The PCB is controlled by the digital control system (DCS) which is operational (only one DCS is powered up at any time). The active DCS is responsible for asserting the address, data, and control lines of the PCB to perform read and write functions on the electronic modules which are attached to the PCB. The PCB distributes all power control and digital interfaces on the spacecraft. The PCB is implemented with programmable peripheral interfaces (PPI, also known as an 8255); however, analog signals from the TMUX and EPS are sent from those modules directly to an analog input port on the DCS for A/D conversion.

PROTOTYPE HARDWARE AND SUPPORT SOFTWARE

Some early prototypes of PANSAT electronic modules were developed before the DCS existed. These modules were the temperature multiplexing system (TMUX), mass storage unit (MSU), and the electrical power system (EPS). These prototypes began as wire-wrapped modules with the appropriate PCB hardware to allow a common interface which was functionally identical to the final design. Early PCB control was implemented using a National Instruments ATMIO-16 A/D and digital control card available for a PC.

The ATMIO-16 card is capable of accepting multiple analog inputs (for A/D conversion) and has four programmable digital lines which can be used for input and output. Analog data from the TMUX and EPS were connected to this board and allowed simple A/D conversion using National Instrument's LabVIEW® programming system. The digital I/O was used to control the PCB. A special circuit was implemented to allow the ATMIO-16 digital I/O bus to emulate controlling the PCB. Software within LabVIEW was created to allow PCB read and write operations. Power for these modules was provided with a programmable power supply which was connected to the General Purpose Interface Bus (GPIB). The PC with the ATMIO-16 also had a GPIB controller. LabVIEW interfaces were used to control the Hewlett Packard 6653A system DC programmable power supply. The first PANSAT prototype was controlled and tested via a PC running LabVIEW with simple Virtual Instruments (VI) allowing discrete PCB write and read operations, conversion and display of A/D, and the ability to turn on and off the modules via the power supply. This configuration is demonstrated with Figure 3.

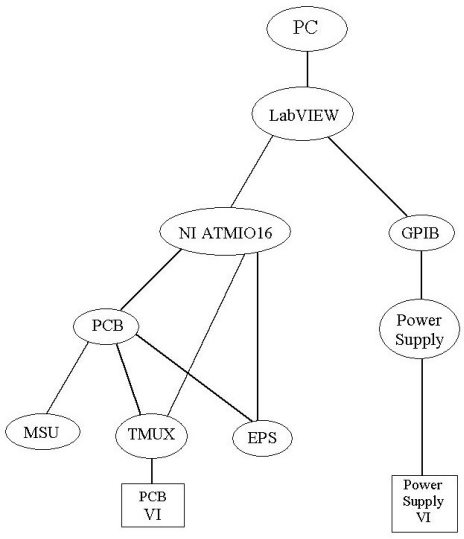

Figure 3. Prototype Support Hierarchy.

LabVIEW was chosen because it is an easy to use and powerful programming tool. It uses a graphical programming language in the form of block diagrams to depict the programming of a system. In addition, LabVIEW offers a rich set of building blocks which makes it very easy to build graphical user interfaces without dealing with the issues of programming these interfaces (Ref. 1). Such a program is called a Virtual Instrument (VI) which consists of a graphical front panel (i.e. the front panel of an instrument which is the interface for a user) and a block

diagram (i.e. the internals of the instrument). Although most of the programming was performed by the author, LabVIEW allowed other engineers to comprehend more easily the structure of the algorithms that controlled the PANSAT modules. Thus, the translation from the engineer's need for control and the implementation were similar and allowed more interaction between the computer programmer and the engineers. In addition, once the algorithms were developed, because the LabVIEW VIs documented clearly the design of the program to control the electronic modules, porting of the VIs to the C programming language on the embedded system of the spacecraft was simplified.

PROTOTYPE SOFTWARE

The prototype hardware described above was exercised and tested using the simple PCB VI which allowed only read and write operations on the PCB. This was an awkward interface for the user because most logical commands or response read-back from the electronic modules require multiple write and read operations on the PCB. This lead naturally to the development of a middle layer of VIs which allowed subsystem commands and display of results to occur within a logical VI for each module.

A mass storage unit VI was created which, using the PCB VI, allowed an address (or a range of addresses) to be specified, the ability to specify read or write, and either the data to write to or a display with the results of reading back data. Thus, writing single bytes, blocks of bytes, or even downloading a file from the PC to the MSU became a very simple operation. Furthermore, automated read and write tests were developed which ran exhaustive tests on the storage units over many days.

Similarly a temperature multiplexing (TMUX) VI was created to allow a thermistor to be specified for conversion and display of its temperature. The selected thermistor caused the TMUX to send an analog signal to the ATMIO-16 A/D board which handled the conversion. Additionally, multiple temperature channels could be selected and displayed on a graph. Testing of the TMUX was simplified by running temperature acquisition over long periods of time in parallel with other laboratory temperature acquisition systems so that comparisons could be made.

The VI for the electrical power system (EPS) allowed the individual power lines of all the electronic modules to be controlled. An interface for the prototype batteries was also developed which allowed the control of the battery switches (trickle charge, full charge, online, and discharge). As PANSAT has two batteries, the battery control was duplicated to allow both batteries to be evaluated. Feedback from the power system (e.g. battery temperatures, voltages, and currents) was controlled via the PCB; the analog results were sent to the ATMIO-16 analog inputs. The direction of battery currents (i.e. *to* or *from* the battery) was encoded by reading a single bit back from the PCB (i.e. the analog signal is only a magnitude).

PROTOTYPE SOFTWARE EVOLUTION

The middle layer VIs described above simplified the control of each individual electronic module of the spacecraft. However, in order to implement actual spacecraft algorithms to perform automated control of the spacecraft subsystems, additional VIs were created.

PANSAT requires continuous power system control. This control is now implemented in ROM on the digital computer on board the spacecraft. However, the first battery control algorithms were developed before the spacecraft existed. These algorithms were created as VIs. The algorithm which correctly decides which battery needs charging and which battery is able to supply online power to the spacecraft is known as the Battery Charge Monitor (BCM). The BCM is a non-trivial algorithm since it must keep track of charge state history of each battery, modify charging and discharging depending on voltages and temperatures of battery cells, and always make sure the preferred battery is online in sunlight and eclipse (Ref. 2). As many anomalies can occur and are only fully identified through testing, implementation of the first BCM was simplified using LabVIEW.

Thorough testing of the batteries and the power system required a more sophisticated power system user interface. Since the solar panels for PANSAT had not yet been acquired, nor was it practical to have the solar panels on the test bench, solar input was simulated using a programmable power supply. An additional VI was created which controlled the power supply while attempting to simulate the orbit of the spacecraft. The solar simulator VI incorporates many variables: the tumbling of the spacecraft (which creates a variable power source since the solar

panels produce different energy levels depending on their orientation with the Sun), orbit time, and Sun versus eclipse times. The front panel of the simulator VI is shown in Figure 4.

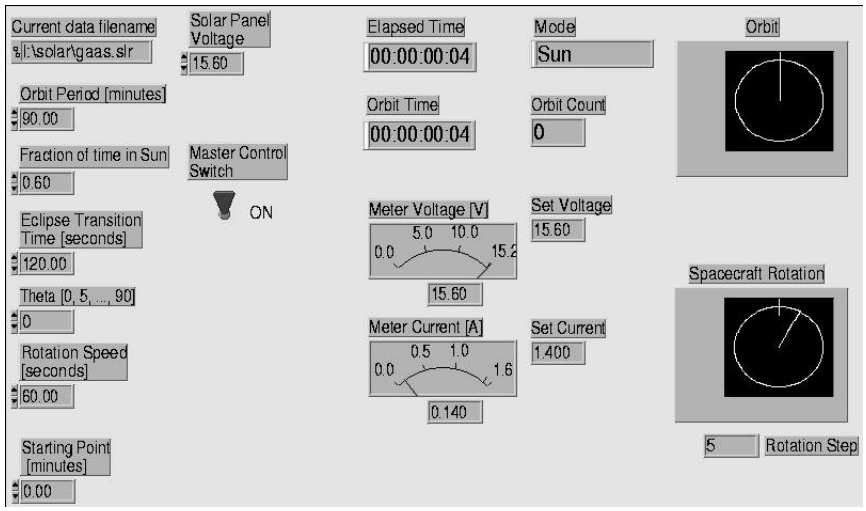

Figure 4. Solar Simulator VI.

With the development of the BCM and Solar Simulator, detailed testing of the PANSAT power system was possible. During this time a power system deficiency was identified. It was discovered that the solar panels would not be able to provide enough power to fully charge the batteries since a higher solar panel voltage (above that of the charging batteries) is required. The batteries were redesigned with one less cell so that over charging of the cells could occur. Figure 5 shows the hardware and software VI hierarchy that existed during this portion of PANSAT's development.

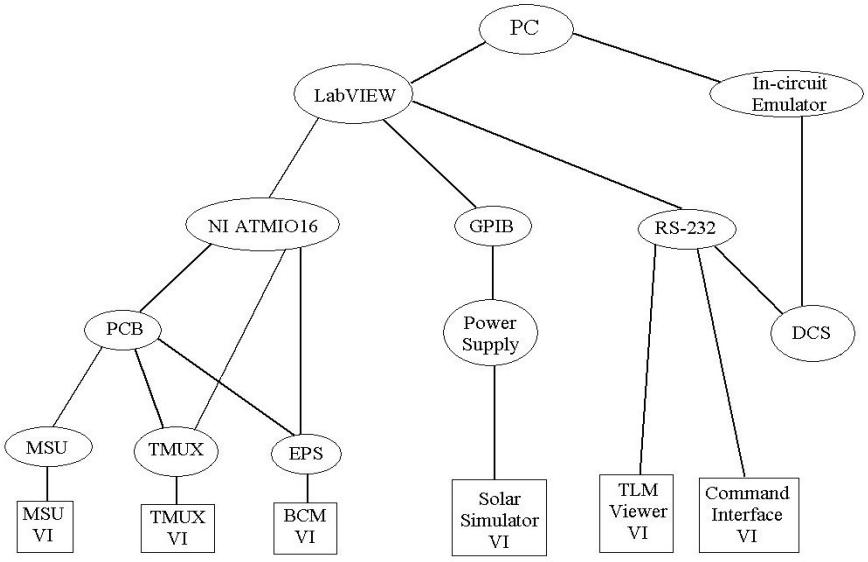

Figure 5. Battery Charge Monitor Support Hierarchy.

SPACECRAFT HARDWARE

The prototype TMUX, MSU, and EPS electronics were implemented as spacecraft flight hardware and environmental testing was performed on these modules. During this time the prototype digital computer, modem, and RF systems were created and tested. The digital computer and the communications hardware were developed

with an in-circuit emulator (ICE box) and RF equipment, separate from the infrastructure used to develop the TMUX, MSU, and EPS.

Eventually, the digital computer replaced the ATMIO-16 A/D and digital I/O board. The DCS has a PPI which is used to control the PCB. In addition, the DCS has an A/D converter which receives signals from the TMUX and EPS. The newly developed DCS was added to the other electronic modules on the test bench in order to test the spacecraft's flight hardware.

SPACECRAFT SOFTWARE

In order to take advantage of the VIs developed during the prototype hardware stage, a software interface was created which allowed the DCS to communicate to a PC using a RS-232 port. Thus, command and control to the spacecraft from the PC was implemented by passing commands and data over a serial port. On the PC, the LabVIEW VIs were modified to allow the PCB operations and A/D conversion to occur on the spacecraft's computer, the DCS. The LabVIEW VIs were modified into modules which displayed data. For the PC to command the spacecraft, another VI was developed which allowed controls to be given on the PC and sent via the serial port to the spacecraft computer.

Because the BCM algorithm was mature when the DCS was created, porting of this algorithm from a LabVIEW VI on the PC to a C program on the spacecraft computer was a straightforward task. Thus, the DCS now automated the spacecraft power system control and the PC was used to monitor the data. Figure 6 shows these hardware and software systems.

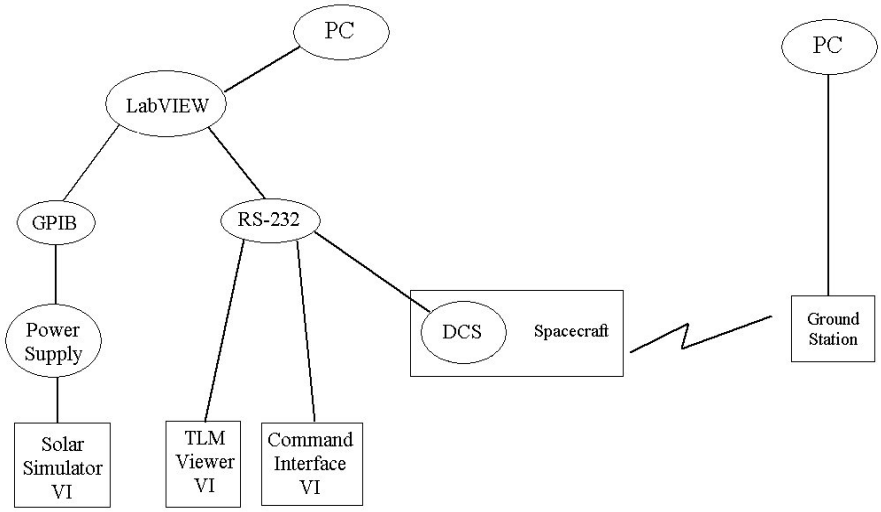

Figure 6. Integration Support Hierarchy.

SPACECRAFT INTEGRATION

At the time of spacecraft integration, all of the electronic modules had been functionally tested independently to insure operation with respect to reliability and workmanship. In addition, module level testing was performed to insure verification of compliance with Shuttle safety requirements (this included random vibration and thermal cycling). A detailed description of the testing requirements is given by Overstreet (Ref. 3).

The ground support hardware used to perform the integration testing at NASA/GSFC consisted of two laptop computers, a Hewlett Packard 6653A DC programmable power supply, and a brief-case sized RF/Modem unit. Using the spacecraft test port interface (STPI) the laptops could exercise each test required to ensure that PANSAT was functioning correctly. The tests performed were as follows.

First a spacecraft computer was powered on. ROM-based software completed an automated hardware check of the computer board to insure all of the hardware peripherals were reset and responded correctly to initialization commands. All of the computer's RAM were checked using moving ones write and read tests (Ref. 4).

Then, the peripherals were powered on and subsystem tests were performed. This required about 15 seconds. The power system was then tested by toggling every solid state switch on the electrical power subsystem. Responses of the switch settings were verified by checking the power supply outputs (voltage and current) and also by viewing the LabVIEW Monitor VI, Figure 7, which shows battery voltages, currents, and temperatures. The power system tests took about two minutes to complete. The mass storage tests, using moving ones write and read techniques, required the most time (about 50 minutes); together, the two memory systems contain 9 Megabytes of random access memory. The temperature multiplexing systems were checked in about one minute using a special command which displayed all thermistor readings on one window. Finally, the communications systems (modem and RF) were tested. The communications systems were designed with certain redundancy such that for each operating computer there are eight unique RF states. The separate RF/Modem unit allowed wireless testing of the spacecraft in the lowest RF power mode at the integration site. The eight communication states were tested in about five minutes. An entire spacecraft system level suite of tests was performed in about one hour. This allowed regular testing to be performed throughout the entire integration process.

Test data were filed after each test. All the graphical windows displaying results of various tests were printed to disk. All command and response data were converted into ASCII-readable output and saved to disk.

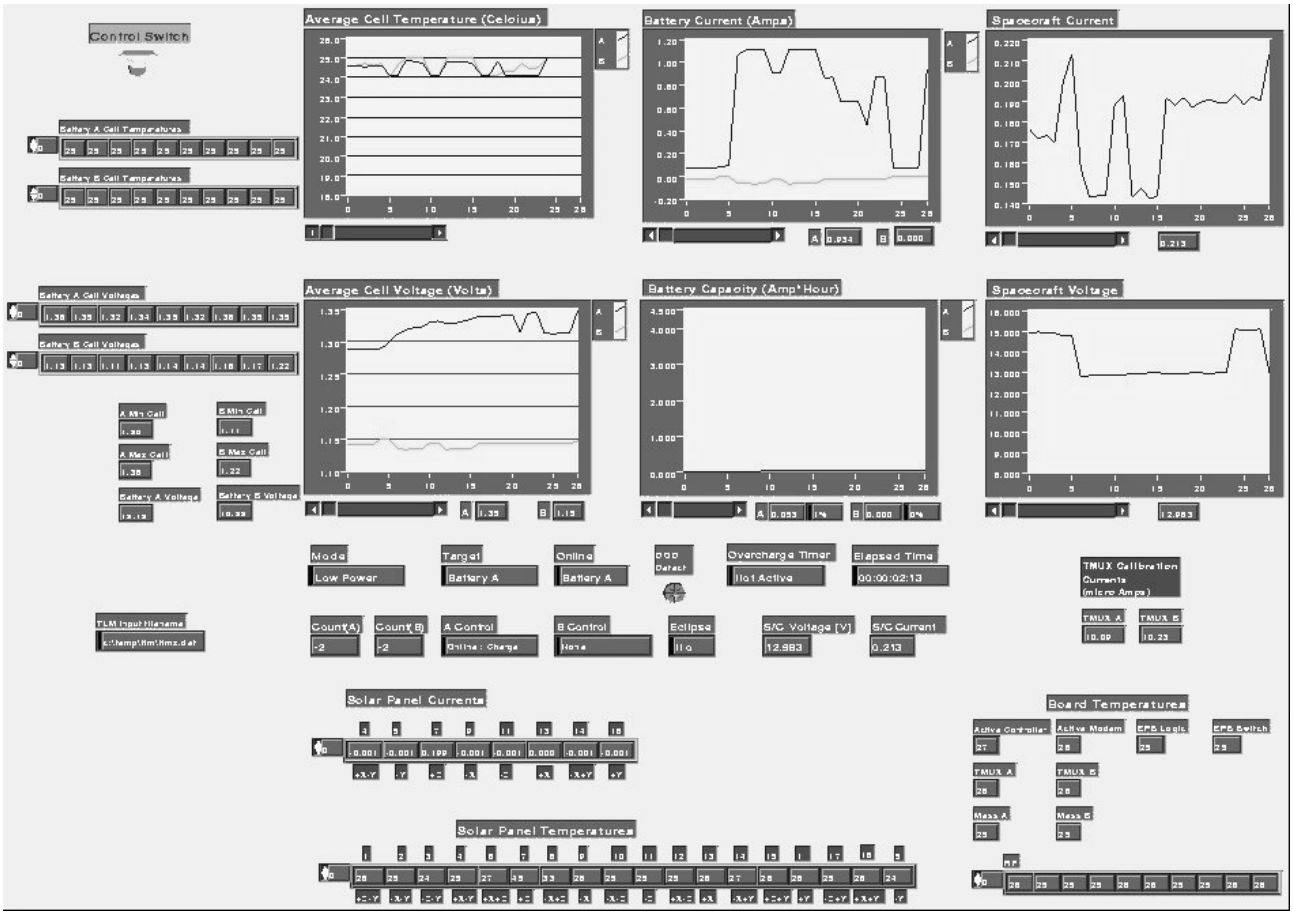

Figure 7. Power System Integration VI.

CONCLUSION

The launch and operation of PANSAT was the effort of several years of work by officer students, staff, and faculty at the Naval Postgraduate School. Successful completion of testing, integrating, and operating an autonomous spacecraft can be accomplished with reusable software modules using a PC, LabVIEW, and some custom programming. The software tools are simple to learn and are appropriate for student instruction.

REFERENCES

1. *LabVIEW User Manual*, Part Number 32099A-01, National Instruments, January 1996.

2. *PANSAT Prototype Batteries And Test Results*, Frank H. Strewinsky, Research Report, Naval Postgraduate School, Monterey, CA, 1996.

3. *Environmental Testing of the Petite Amateur Navy Satellite (PANSAT)*, Paul J. Overstreet, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1997.

4. *Programming Embedded Systems in C and C++*, Michael Barr, O'Reilly and Associates, January 1999.